# O-Tech Computer Shop

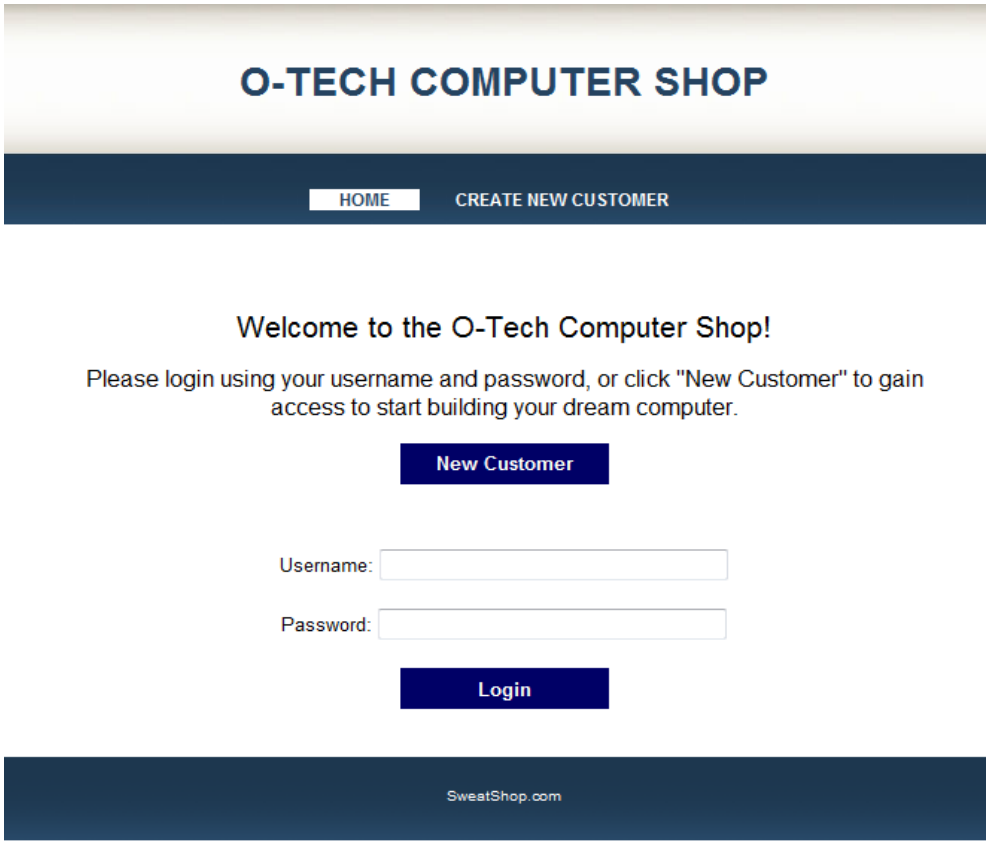
**TEAM 4**
**Jorge Ortiz, Karl King, and Dan Ehlers**
**Friday, December 03, 2010**

# Overview

As Group 4, we felt responsible to do our absolute best to uphold the reputation of the prestigious group number that previous semester's Group 4s have built. We first were going build a website in which a user could access to build a custom airplane or jet. However, once we began to gather information on the topic, some of us realized that we know little to nothing about the components in which it takes to build an airplane. Thus, we decided to go with an idea for which we know much about, computers.

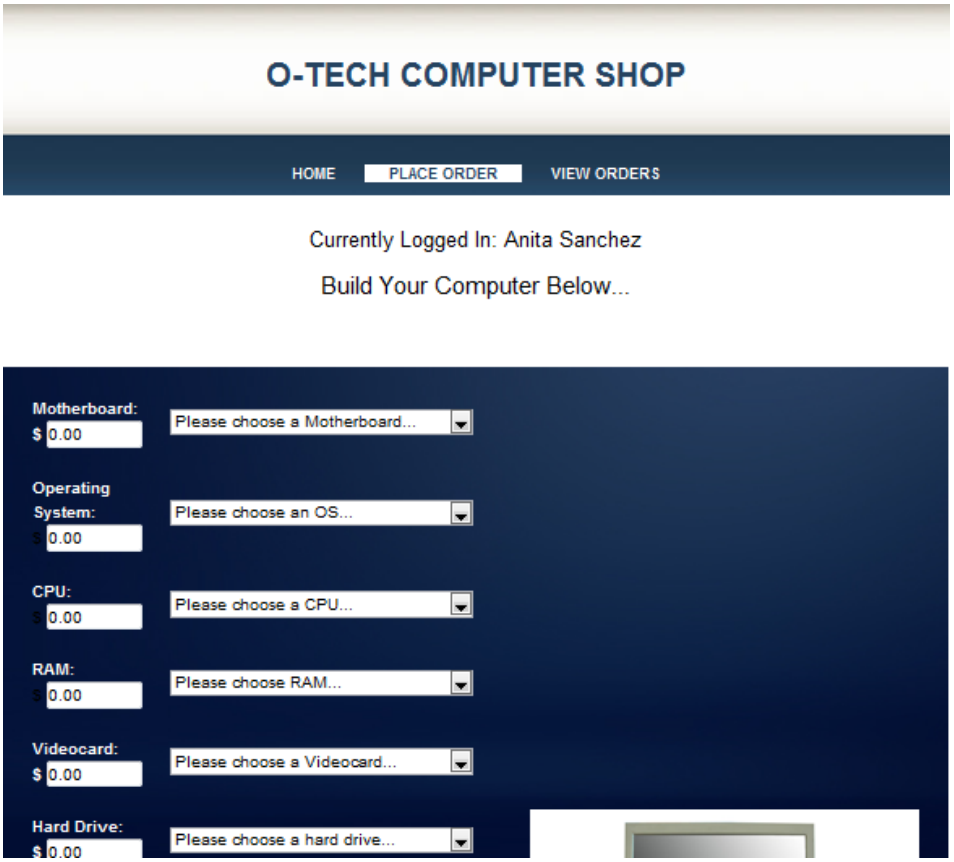
We then began gathering information and planning on a website in which a customer could access to build a custom computer. Quickly, the idea was derived where it would be ideal to create a system for which an entire business could be run from the website. Therefore, rather than having a business dependent on the computers for which a software program would be installed and used, the business could function anywhere there is an internet connection and a way to access the website.

The website allows for multiple types of users to use the system, and, depending on the user, different functionalities would allow the user to accomplish whatever task they need to do.



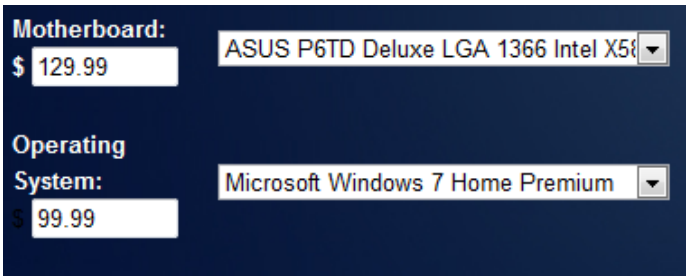
# Client Functionality

As previously stated, this website was designed to be an all-purpose custom computer website for which all aspects of the business can be managed. There are four different types of personnel for which the system is designed for. The first type of person which can use the system is the all-too-important customer. The customer has limited functionality, as you will soon see, when compared to other users. To gain access as a customer, from the homepage, there is a convenient link for which a new customer can sign up. The sign up page has a series of error checking algorithms to ensure that the information entered is correct. Such examples are comparing that entered passwords are the same, email confirmation, zip code accuracy, and state legibility. Once the customer is registered, he or she will be added to the database where the information is stored for later use. Above is a screenshot of the homepage login screen. Once the customer is logged in, he/she will be directed to the order page where orders can be placed.



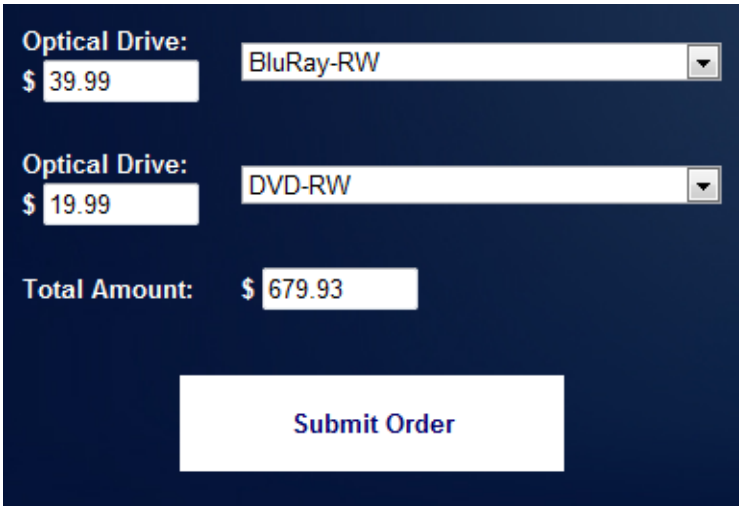

Above you can see a screenshot of the customer order page, order.aspx. You will notice a confirmation of the logged in user at the top of the page to ensure accuracy and reduce mistakes. Next, is a series of dropdown menus which holds a variety of different software and hardware products. Each menu is populated upon the loading of the page using the data in the

Products table. Once the user selects a specific item from a dropdown menu, the item's price is displayed in a box to the left. This is shown in the screen shot below.




Another feature of the order page is whenever a new item is selected from one of the dropdown menus, a URL corresponding to the newly selected item is retrieved from the database and its picture is loaded to the right of the list of menus. A screenshot of this can be seen below.

The last feature of the order page is a continuous tabulation of all of the selected items' monetary value which is displayed in a box labeled "Total Amount." Again, this is shown in a screenshot below.



Once the user is satisfied with their order, they can submit the order by clicking the "Submit Order" button (shown above). Once clicked, the order is added to the database where it can be viewed later by employees. The employee is then directed to the a page where they can review their order. This page is also the only page that the second type of user, Warehouse Employee, has access to. Since they will only need to know what the components of the computer for which they will be building and shipping, this is the only required privilege.

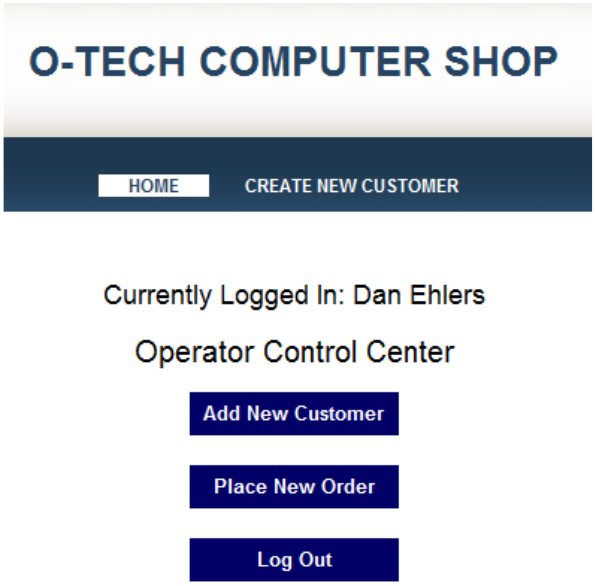
View Order

The view all page will read all the order ids and add them to the drop list show on the above screenshot. When the user select one of the order ids, the program will populate each text box as read only. It will specify the motherboard, operating system, CPU, video card, hard drive, peripherals, keyboard language, and optical drive related to that order id. An example of this it is show on the fallowing screenshot.

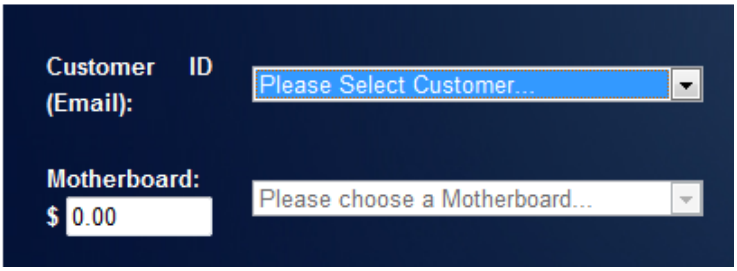
Order ID:
975112802
Motherboard:
EVGA P55 LE 123-LF-E653-KR LGA1156 Intel P55 ATX Intel
Operating System:
Ubuntu v10.10
CPU:
AMD Phenom II X6 1055T Thuban 2.8GHz 6 x 512KB L2 Cache 6MB L3 Cache Socket AM3 125W Six-Core Desktop Processor HDT55TFBGRBOX
Videocard:
PNY VCGGTX580XPB GeForce GTX 580 (Fermi) 1536MB 384-bit GDDR5 PCI Express 2.0 x16 HDCP Ready SLI Support Video Card
Hard Drive:
Western Digital RE3 WD1002FBYS 1TB 7200 RPM 32MB Cache SATA 3.0Gb/s 3.5 Inch Internal Hard Drive -Bare Drive
Hard Drive:
Western Digital RE3 WD1002FBYS 1TB 7200 RPM 32MB Cache SATA 3.0Gb/s 3.5 Inch Internal Hard Drive -Bare Drive
Hard Drive:
Western Digital RE3 WD1002FBYS 1TB 7200 RPM 32MB Cache SATA 3.0Gb/s 3.5 Inch Internal Hard Drive -Bare Drive
Hard Drive:
None
Peripherals:
USB 3.0 No
SATA No
Firewire No
Bluetooth No
Keyboard Language:
English
Optical Drive:
DVD-RW
Optical Drive:
DVD-RW

The third type of user that can login to the system is an operator. The operator is intended to only answer calls from customers. Once the operator logs in, they will see an

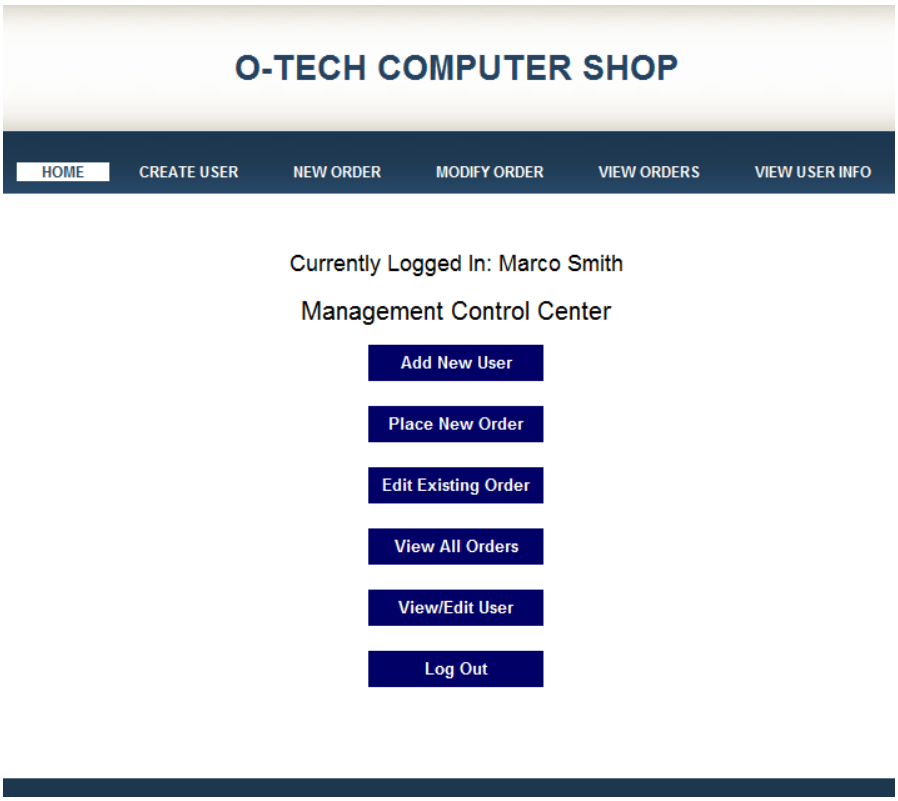
Operator Control Center page (pictured below) which shows the different options for which they have access to.




Therefore, the privileges of the operator are restricted to only creating new customer accounts and placing orders. The only difference in the order page for the Operator Order Page is that the operator needs to choose which customer account the order will be charged to. The ordering menu and order submission button are disabled until the operator selected a customer. A screenshot of this is shown below.

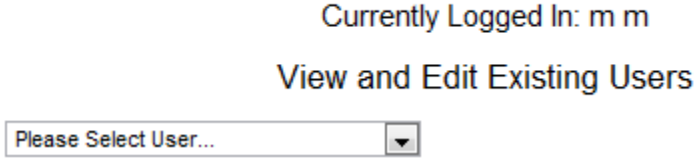



The fourth and final type of use that has access to the system is a member of Management. Once a manager has logged into the system, a Management Control Center is displayed which shows all of the available options in the system.

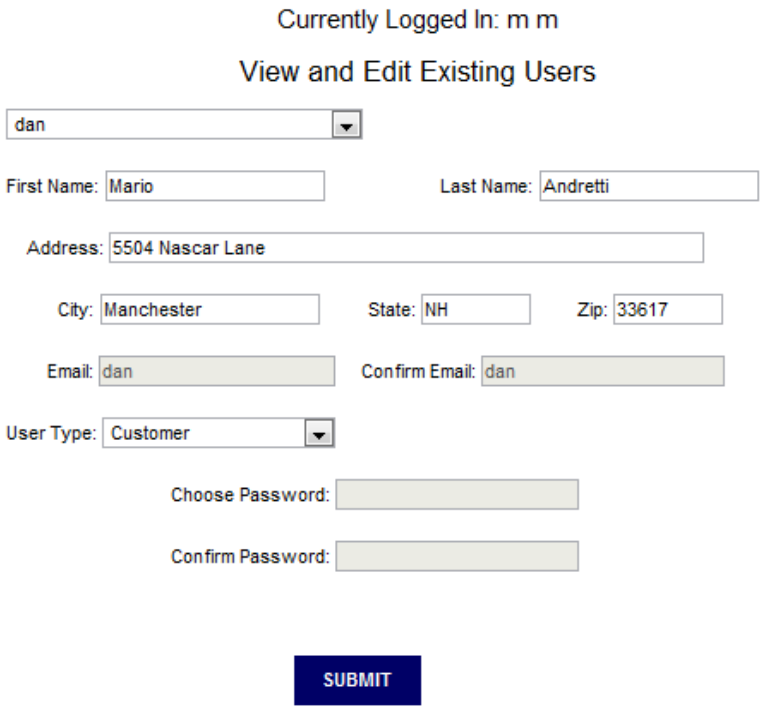



Since managers need to do various tasks throughout the day such as add new employees, update user information, or edit an existing order, they have some extra functionality that operators do not (and should not) have access to. One such page is the Edit Existing Order page which allows the manager to choose a customer ID from a dropdown menu. Once an ID is selected, a second dropdown menu is populated with all Order IDs of all of the order placed by that customer. Once the Order ID desired is selected, the order form populates all of the product menus and automatically selects the appropriate item for each menu according to the order that was selected and consequently pulled from the database. The manager can then make and changes to the order that the customer specifies. Once finished, the manager can submit the order, but rather than simply adding the order to the database, the order will simply update the existing order in the database. A screenshot of the Edit Order Page is shown below.

The only user types that have access to the view user info are the managers. When a manager logs to the program and goes to the view user info page the program will read all the user ids for the Members table. After it read all the user ids they will be added to the drop list show on the following screenshot.



After the manager select one of the user ids for the drop list the program will populate each text box with the information related with that user. This option give the manager the ability to do changes to user first name, last name, address, city, state, zip, and user type. The email and password text box cannot be modified by the manager. On the following screenshot we can see an example of this.




After the changer are done to the user account the manager will proceed by clicking on the submit button. At this stage the program will update the Members table with the new changers for that specific user.

Currently Logged In: Marco Smith

Modify Existing Order Page

# Database Design and Tables:

## a) Products:

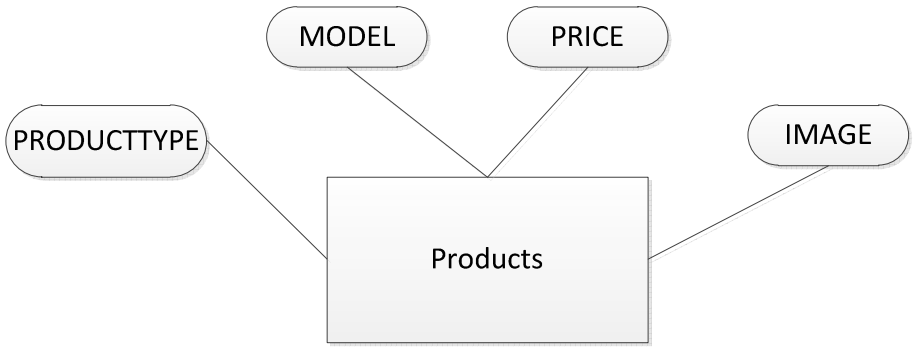



**Figure 1: Products Schema**

Products(<u>PRODUCTTYPE</u>, MODEL, PRICE, IMAGE);

Foreign keys: None
Primary key: PRODUCTTYPE
Not null: PRODUCTTYPE

The above diagram illustrates the schema of the Products table. This table holds the information of all of the different products that can be used in the creation of the custom computer. The attribute PRODUCTTYPE tells what type of product is in that row which allows us to create a website that can distinguish between different types of hardware and software components. The MODEL attribute holds the necessary information needed to distinguish between different products of the same PRODUCTTYPE. The PRICE attribute holds the value for which the website will charge for that particular item. Lastly, the IMAGE column holds the URL for which a picture of that product is located so that the website can display an image for a potential buyer. Sample data from the Products table is located below.

| TYPE | MODEL | PRICE | IMAGE |
|---|---|---|---|
| Motherboard | ASUS P6TD Deluxe LGA 1366 Intel X58 ATX Intel | 229.99 | /images/ |
| Motherboard | EVGA P55 LE 123-LF-E653-KR LGA1156 Intel P55 ATX Intel | 169.99 | /images/ |
| OS | Microsoft Windows 7 Home Premium | 99.99 | /images/ |
| OS | Apple Mac OS X v10.6 | 29.00 | |
| OS | Ubuntu v10.10 | 0.00 | /images/ |
| OS | Mcrosoft Windows 7 Ultimate | 179.99 | |
| CPU | i7-875K Lynnfield 2.93GHz 8MB L3 Cache LGA 1156 95W Quad-Core Unlocked Desktop Processor BX80 | 329.99 | /images/ |
| CPU | 1055T Thuban 2.8GHz 6 x 512KB L2 Cache 6MB L3 Cache Socket AM3 125W Six-Core Desktop Processo | 179.99 | /images/ |
| VideoCard | 80XPB GeForce GTX 580 (Fermi) 1536MB 384-bit GDDR5 PCI Express 2.0 x16 HDCP Ready SLI Supp | 529.99 | /images/ |
| VideoCard | eon HD 5870 (Cypress XT) 2GB 256-bit GDDR5 PCI Express 2.1 x16 HDCP Ready CrossFireX Support Eyefi | 379.99 | |
| HDD | agate Barracuda LP ST32000542AS 2TB 5900 RPM 32MB Cache SATA 3.0Gb/s 3.5" Hard Drive -Bare Dri | 99.99 | /images/ |
| HDD | tern Digital RE3 WD1002FBYS 1TB 7200 RPM 32MB Cache SATA 3.0Gb/s 3.5" Internal Hard Drive -Bare I | 129.99 | /images/ |
| USB3 | | 24.99 | |
| SATA | | 30.00 | |
| FIREWIRE | | 40.00 | |
| BLUETOOTH | | 15.00 | |
| KEYBOARDTYPE | | 19.00 | |
| ODD1 | DVD-RW | 19.99 | |
| ODD2 | BluRay-RW | 39.99 | |

## b) Order:

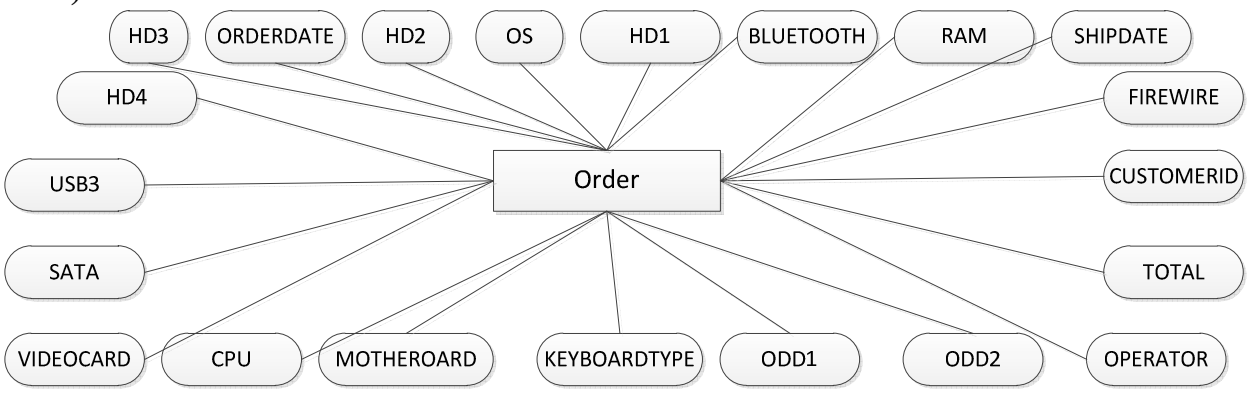



**Figure 2: Order Schema**

Order(CUSTOMERID, ORDERID, MOTHEROARD, OS, CPU, RAM, VIDEOCARD, HD1, HD2, HD3, HD4, USB3, SATA, FIREWIRE, BLUETOOTH, KEYBOARDTYPE, ODD1, ODD2, OPERATOR, TOTAL, SHIPDATE, ORDERDATE);

Foreign keys: CUSTOMERID
Primary key: ORDERID
Not null: CUSTOMERID, ORDERID, TOTAL, SHIPDATE, ORDERDATE

At the heart of the website lies the Order table. This table holds all necessary information for the company in order to successfully build the custom computer. Not only does it store all of the different components for which the customer has selected to be included in the creation of their custom computer, but it also keeps the necessary information to ensure quality control and accountability. If a customer had called the business and spoke to a customer care representative to place an order, the system will keep track of the operator who placed the order, as well as the date and time of the order. Similarly, once an order is shipped, the warehouse personnel can mark it as shipped and the program will automatically add the date and time of the shipment.

## c) Members:

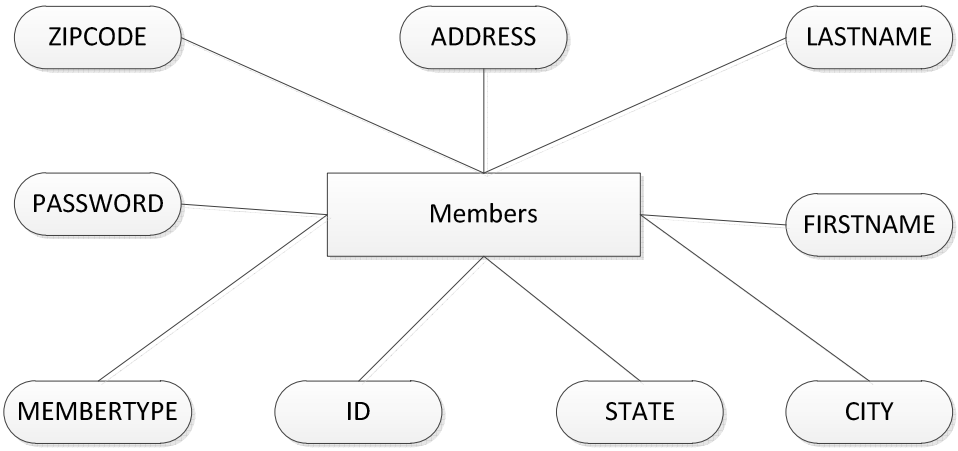



**Figure 3: Member Schema**

Members(ID: varchar(50) , FIRSTNAME: varchar(50) , LASTNAME: varchar(50) , ADDRESS: varchar(50) , CITY: varchar(50) , STATE: varchar(3) , ZIPCODE: int , PASSWORD: varchar(50) , MEMBERTYPE: varchar(50) );

Foreign keys: ID
Primary key: ID
Not null: ID, FIRSTNAME, LASTNAME, ADDRESS, CITY, STATE, ZIPCODE, PASSWORD, MEMBERTYPE

The last of the three tables is the Members table. This table holds all of the information needed about a user of the system. In order for a user to gain access to the system, he or she must be registered. The data in this table helps the system determine which privileges the current user has. Some sample data held in the Members table can be found below.

| ID | FIRSTNAME | LASTNAME | ADDRESS | CITY | STATE | ZIPCODE | PASSWORD |
|---|---|---|---|---|---|---|---|
| tellus@Suspendisse.org | Aurelia | Veda | P.O. Box 808, 3895 Phasellus Avenue | Huntington Park | KS | 94255 | sit |
| magna@iaculis.com | Jeremy | Sybil | 912-1339 Felis St. | Muskogee | ME | 91599 | enim |
| ullamcorper@felisNulla.edu | Thomas | Harriet | Ap #220-6297 Fermentum Ave | Akron | VA | 37556 | bibendum |
| eger@porttitorvulputateposuere.e | Naomi | Celeste | P.O. Box 231, 3075 Rhoncus. Avenue | Kona | TX | 4375 | luctus |
| mattis.velit@nunc.org | Alan | Nomlanga | 405-5092 Mus. Street | Gallup | MA | 64186 | purus. |
| dui.nec@duiquisaccumsan.org | Jennifer | Emma | P.O. Box 427, 8848 Et, St. | Narragansett | IN | 96629 | facilisis, |
| lectus.a@Nullaegetmetus.ca | Charles | Lavinia | 541-1847 Lacus. Av. | Victoria | OK | 34548 | dapibus |
| sollicitudin@iaculisodio.edu | Irene | Barry | 3201 Ante. Road | Valparaiso | LA | 35720 | nibh. |
| odio.auctor.vitae@eratvolutpat.con | Devin | Jack | 8797 Arcu Avenue | Hastings | UT | 30844 | velit |
| sagittis.Duis@duiCras.org | Remedios | Quamar | 1465 Ut Av. | Cerritos | IN | 85494 | ornare |
| us.Donec.egestas@orciquislectus.c | Colton | Raphael | 3163 Nulla. Road | Cortland | NY | 17916 | ut |
| Nulla.eget.metus@lectusante.ca | Petra | Gage | 4195 Odio, Road | Missoula | NH | 38022 | venenatis |
| sit.amet@dui.org | Xanthus | Kylan | 6508 Velit Ave | La Mirada | IL | 73377 | Praesent |
| aliquet@Duisrisus.edu | Cherokee | Breanna | 458-428 Maecenas St. | West Jordan | TN | 63132 | malesuada |
| nostra.per@iaculis.org | Rose | Reuben | 891-1224 Commodo Avenue | Palmdale | NC | 76710 | Aliquam |
| velit.Quisque.varius@in.ca | Patrick | Callum | P.O. Box 378, 6101 Lectus Av. | New Rochelle | NH | 25954 | luctus |
| Aenean@acsemut.edu | Alexandra | Dillon | P.O. Box 434, 2712 Malesuada St. | Wynne | PA | 50253 | eget |
| a.magna.Lorem@sem.com | Sybil | Cherokee | 581 Etiam Av. | Auburn Hills | WV | 86597 | tincidunt |
| am.dignissim@PhasellusornareFus | Wilma | Delilah | 3292 Velit Rd. | Highland Park | WY | 4512 | urna |
| tristique@diamluctuslobortis.edu | Hamish | Cassandra | P.O. Box 215, 8050 Ultricies Av. | La Habra Heights | VT | 17265 | In |
| et.malesuada@pede.com | Kylan | Alfonso | Ap #203-9324 Blandit Ave | Taunton | MI | 69981 | ullamcorper |
| lectus@neccursus.edu | Raymond | Fritz | Ap #493-8146 Phasellus Av. | Cedar Rapids | IN | 67518 | Proin |
| odio@tellus.edu | Jerry | August | P.O. Box 991, 1905 Risus. Av. | Dayton | IA | 37191 | leo. |
| idio.auctor@quamelementumat.or | Wynne | Beverly | Ap #924-3202 Vel Street | Torrington | AK | 17737 | nunc. |
| Donec@urnaVivamusmolestie.com | Tobias | Kylynn | Ap #137-4417 Integer Street | Ypsilanti | AL | 77236 | elit |
| ullamcorper.magna.Sed@augue.ca | Leila | Cora | Ap #765-7482 Elit, St. | Murfreesboro | MS | 97347 | quis, |
| itant.morbi.tristique@egestasnunc | Owen | Jena | Ap #431-1017 Dolor Av. | Daly City | MS | 30297 | egestas |
| sed@estmauris.com | Yvette | Juliet | P.O. Box 560, 994 Odio. Rd. | Anaconda | NH | 43497 | nulla. |
| ante@commodo.org | Keith | Brennan | Ap #460-9828 Integer Rd. | Reedsport | AK | 42467 | pellentesque |
| in@ac.org | Leonard | Nicholas | 319-7485 Urna. Rd. | Miami Gardens | WI | 73627 | at |
| .lobortis@MorbivehiculaPellentesc | Lev | Xanthus | P.O. Box 947, 6636 Rutrum Avenue | Glen Cove | CT | 76608 | vestibulum |
| lit.Pellentesque@porttitortellusno | Ila | Leonard | 935-7698 Arcu Av. | Detroit | WI | 29429 | tellus. |
| aliquam@elitsed.org | Brett | Lucy | Ap #788-3726 Velit Ave | Kearns | CO | 84823 | Proin |
| lobortis@anteNuncmauris.edu | Leigh | Jane | P.O. Box 598, 4725 Dictum Av. | Cairo | NM | 34682 | est |
| nunc.est@aliquameros.org | Aimee | Gannon | 7610 Odio. Road | South El Monte | GA | 71262 | risus. |
| sollicitudin.orci@dui.org | Rhiannon | Blake | Ap #955-8889 Nunc Av. | West Allis | IA | 45483 | sed, |
| liquam.adipiscing@egestasligula.co | Daphne | Myles | Ap #592-1394 Elit St. | Troy | WV | 60920 | dolor |
| amet.metus@tellussem.com | Summer | Dalton | 2700 Vestibulum Rd. | Fulton | ME | 78573 | dolor, |
| fames@erat.org | Bradley | Uma | P.O. Box 465, 2554 Urna. Road | Easton | OR | 55812 | ligula. |
| justo.Praesent.luctus@nulla.com | Ava | Jesse | Ap #982-1687 Ultrices. Street | West Warwick | AZ | 78782 | orci |
| ler.cursus.Integer@rutrumeuultrice | Wilma | Armando | 7656 Ridiculus Ave | Charleston | NH | 6323 | quis, |
| lacus@fringillaornareplacerat.edu | Erasmus | Ignatius | 9281 Est. Road | Gilette | AL | 84050 | pharetra |
| quis@ridiculusmus.com | Tamekah | Fleur | Ap #688-3116 Auctor St. | Chicopee | WI | 67707 | ipsum |
| id.magna.et@atiaculisquis.ca | Meghan | Ginger | P.O. Box 692, 5302 Phasellus St. | Escondido | CT | 64931 | sodales |
| dapibus@primisin.org | Diana | Kadeem | 943-1129 Risus. Ave | Danville | MO | 46052 | sed, |
| tae.velit@gravidanonsollicitudin.ec | Leigh | Ray | 974-8692 Est Road | Hermitage | VT | 66806 | Sed |
| lorem@eu.com | Wylie | Ebony | 7869 Turpis. St. | The Dalles | FL | 27350 | netus |
| turpis.Aliquam@blanditatnisi.ca | Eve | Irma | 1734 Nulla Road | Glen Cove | DE | 28649 | dolor |
| massa.non.ante@Fuscemilorem.ca | Idona | Eagan | 601-1596 Placerat. Av. | Sioux City | WI | 41280 | rhoncus |
| Donec@tellusAenean.org | Hedley | Charlotte | Ap #478-2701 Amet, Avenue | Inglewood | AK | 86721 | consectetuer |
| sociosqu@lobortistellusjusto.com | Timon | Baxter | 465-6473 Arcu Rd. | North Pole | CA | 60114 | Curabitur |
| felis.orci@enimcondimentum.edu | Hadley | Lucian | P.O. Box 274, 6647 Orci Rd. | Grand Forks | MN | 84438 | mi |
| ugue.scelerisque.mollis@facilisis.ed | Amy | Yasir | 625-4110 Elit. Av. | Visalia | AK | 2824 | Maecenas |
| t.dolor.dapibus@Suspendisseac.co | Jermaine | Marny | P.O. Box 146, 600 Sit St. | Cohoes | CO | 13569 | ultrices |
| rnare.tortor.at@quamquisdiam.ed | Yvonne | Kyra | Ap #193-125 Vel, Road | Huntington Beach | AK | 35266 | dolor. |
| sed.est.Nunc@idenim.ca | Donna | Roth | Ap #249-1150 Elit, Rd. | Poughkeepsie | CA | 25176 | faucibus |
| .scelerisque.scelerisque@Fusce.co | Cailin | Camille | 7958 Mus. Ave | Macon | OR | 60317 | lacinia |
| libero.Proin@Sed.ca | Jasper | Emerald | Ap #465-5432 Nullam Road | Owensboro | AK | 16486 | ac, |
| us.Donec@consectetueradipiscing.( | Elijah | Halla | 343-4793 Eu St. | Rehoboth Beach | SD | 54308 | risus. |
| n.euismod.mauris@cursusIntegerm | Venus | Quinn | 576-3155 Egestas Avenue | Rye | AK | 56988 | magna. |
| risus.Donec.egestas@nonsapien.or | Hope | Lance | 907-1602 Arcu. Road | Galesburg | CO | 91336 | ipsum |
| it.amet.faucibus@risusquisdiam.ed | Mariko | Bruno | 2498 Feugiat St. | Olean | NY | 98132 | sem |
| mauris.sit@Loremipsum.edu | Barclay | Jade | 231-9736 Molestie Street | Newport | TX | 14314 | Nullam |
| felis@odio.ca | Leslie | Shaine | P.O. Box 417, 7314 Nostra, Street | Philadelphia | NE | 65017 | diam. |
| bibendum.sed@Aenean.ca | Abdul | Rhoda | 335-6979 Amet Av. | Hartford | FL | 613 | nibh |
| ıdum.sed.est@bibendumullamcorp | Genevieve | Rinah | Ap #581-5911 Mauris Av. | Dunkirk | WV | 85718 | vestibulum |
| Sed@etpedeNunc.edu | Ivana | Kiona | Ap #873-9653 Dapibus St. | Missoula | IN | 20995 | Sed |
| sodales.nisi@a.edu | Kyla | Clementine | 3191 In, Rd. | Blacksburg | VA | 97414 | erat. |
| Praesent.eu.dui@estcongue.ca | Gloria | Ayanna | 1018 Eget St. | Pittsfield | CA | 19204 | risus |
| sed.tortor@arcuVestibulumut.ca | Basil | Griffin | P.O. Box 179, 4319 Curabitur St. | City of Industry | AK | 29238 | orci. |
| pharetra.sed.hendrerit@eros.ca | Halee | Barry | 605-5453 Nec St. | Woburn | NV | 51661 | et |
| Proin@nisiCum.edu | Amena | Regan | 1816 Lacinia Avenue | York | UT | 90714 | vulputate |
| turpis.egestas@Lorem.com | Eden | Nita | P.O. Box 506, 6532 Molestie. Rd. | Santa Clara | OK | 28118 | at, |
| nunc.risus.varius@loremsit.com | Emery | Burke | 4061 Vulputate Av. | Medford | LA | 36996 | Maecenas |
| consequat.dolor@euplacerat.com | Hedy | Lois | Ap #124-2070 Primis Avenue | North Platte | OH | 31120 | nulla. |
| Integer.eu@feugiatSed.com | Kim | Phyllis | P.O. Box 808, 3340 Malesuada Ave | Menomonee Falls | VA | 29279 | Nam |
| apibus.quam@convallisconvallis.ed | Imani | Hunter | P.O. Box 232, 9924 Est Street | Normal | WV | 38146 | feugiat. |
| sem.ut@duiin.edu | Ivy | Allegra | P.O. Box 116, 9807 Eu Rd. | Palm Springs | CA | 20941 | placerat |
| vitae@Cumsociis.edu | Veda | Boris | 967 Non, St. | Ashland | KY | 13963 | luctus |
| odio@lacinia.com | Acton | Molly | 3189 Vitae Rd. | Mobile | WA | 49109 | Donec |
| ımet.ornare.lectus@Nullamnisl.con | Willow | Carson | P.O. Box 128, 3906 Arcu Road | Saratoga Springs | AK | 78036 | egestas. |
| ltricies.ornare@elitAliquamauctor.( | Gwendolyn | Warren | Ap #989-2436 Vehicula St. | Plantation | IN | 73483 | magna. |
| dis.parturient@idmollis.edu | Silas | Medge | 172 Lacinia Rd. | Laconia | IA | 35332 | euismod |
| tempus.lorem.fringilla@orci.com | Hanae | Lucas | P.O. Box 964, 4333 Et, Rd. | Thibodaux | LA | 58106 | mi |
| te.dictum.mi@nostraperinceptos.c | Brady | Joseph | 6276 Vitae, Rd. | Dunkirk | RI | 86049 | enim, |
| egestas.urna.justo@luctuslobortis.c | Althea | Stacy | 1840 Montes, Road | Lander | MI | 24950 | rutrum |
| sit@amagna.com | Nayda | Carla | 318 Vitae Av. | Toledo | NC | 20284 | tellus. |
| orci.quis.lectus@nunc.ca | Candice | Zane | 623-9261 Ultricies Street | Morgantown | UT | 67155 | nec |

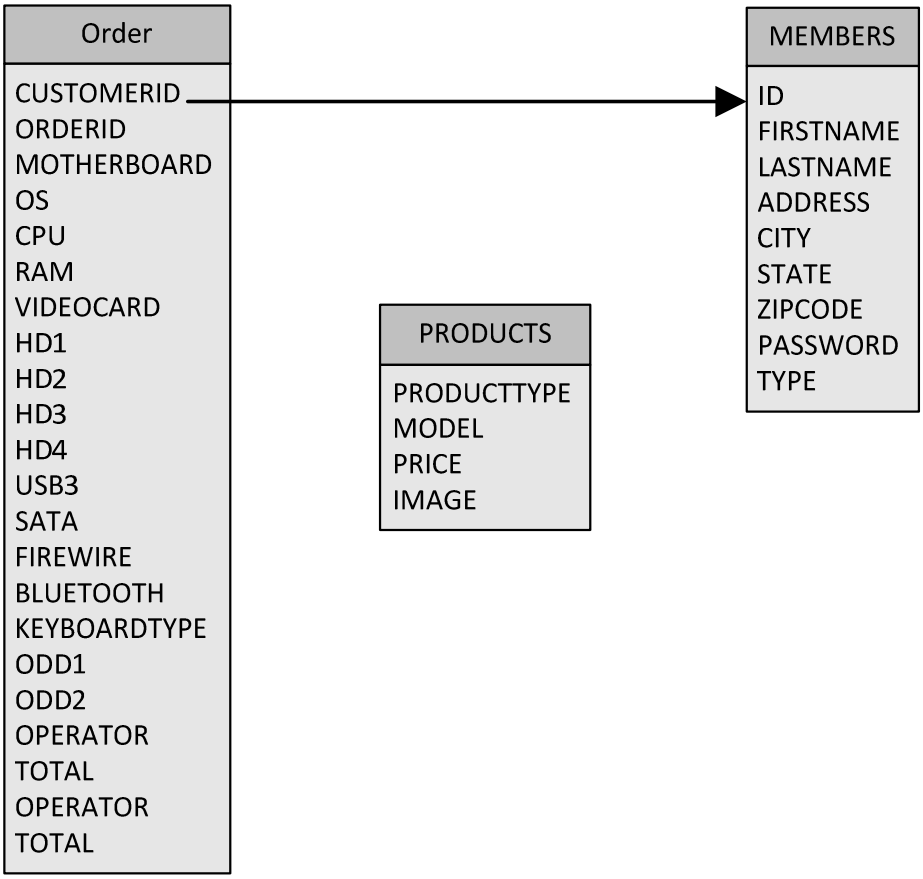



**Figure 4: The diagram above depicts the relationship between the three tables.**

# Database Queries

**1) From customers.cs file, the following SQL insert statement is used to add members to the members table:**

```
"INSERT INTO Members " +
                 " (ID, FIRSTNAME, LASTNAME, ADDRESS, CITY, STATE, ZIPCODE,
PASSWORD, MEMBERTYPE) " +
                 " VALUES " +
                 " ('" + R(ID) +
                    "','" + R(firstName) +
                    "','" + R(lastName) +
                    "','" + R(address) +
                    "','" + R(city) +
                    "','" + R(state) +
                    "','" + zip +
                    "','" + R(password) +
                    "','" + R(memberType) + "')";
```

**2) From orders.cs file, the following SQL insert statement is used to add orders in to the orders table:**

```
"INSERT INTO Orders" +
                  " (CUSTOMERID, ORDERID, MOTHERBOARD, OS, CPU, RAM,
VIDEOCARD, HD1, HD2, HD3, HD4, USB3, SATA, FIREWIRE, BLUETOOTH, KEYBOARDTYPE,
ODD1, ODD2, OPERATOR, TOTAL, SHIPDATE, ORDERDATE) " +
                  " VALUES " +
                  " ('" + R(customerID) +
                  "','" + R(orderID) +
                  "','" + R(motherboard) +
                  "','" + R(os) +
                  "','" + R(cpu) +
                  "','" + R(ram) +
                  "','" + R(videoCard) +
                  "','" + R(hd1) +
                  "','" + R(hd2) +
                  "','" + R(hd3) +
                  "','" + R(hd4) +
                  "','" + R(usb3) +
                  "','" + R(sata) +
                  "','" + R(firewire) +
                  "','" + R(bluetooth) +
                  "','" + R(keyboard) +
                  "','" + R(odd1) +
                  "','" + R(odd2) +
                  "','" + R(operatorID) +
                  "','" + R(total) +
                  "','" + R(shipDate) +
                  "','" + R(orderDate) + "')";
```

**3) From products.cs file, the following SQL statement is used to add the products into the products table:**

```
"INSERT INTO Members " +
                  " (PRODUCTTYPE, MODEL, PRICE, IMAGE) " +
                  " VALUES " +
                  " ('" + R(productType) +
                     "','" + R(model) +
                     "','" + R(price) +
                     "','" + R(image) + "')";
```

**4) From editOrder.aspx.cs file, the following SQL statements is used to get products from the products table:**

```
"SELECT * FROM Products";
```

**5) From editOrder.aspx.cs file, the following SQL statements is used to get a specified member based on order ID products from the members table:**

```
"SELECT * FROM Members " +
"ORDER BY ID";
```

**6) From editOrder.aspx.cs file, the following SQL statement is used to get all the orders by customer ID:**

```
"SELECT * " +
"FROM Orders " +
"ORDER BY CUSTOMERID ";
```

**7) From ViewAll.aspx.cs file, the following SQL statement is used to get all the orders from the order table:**

```
"SELECT *  " +
"FROM Orders ";
```

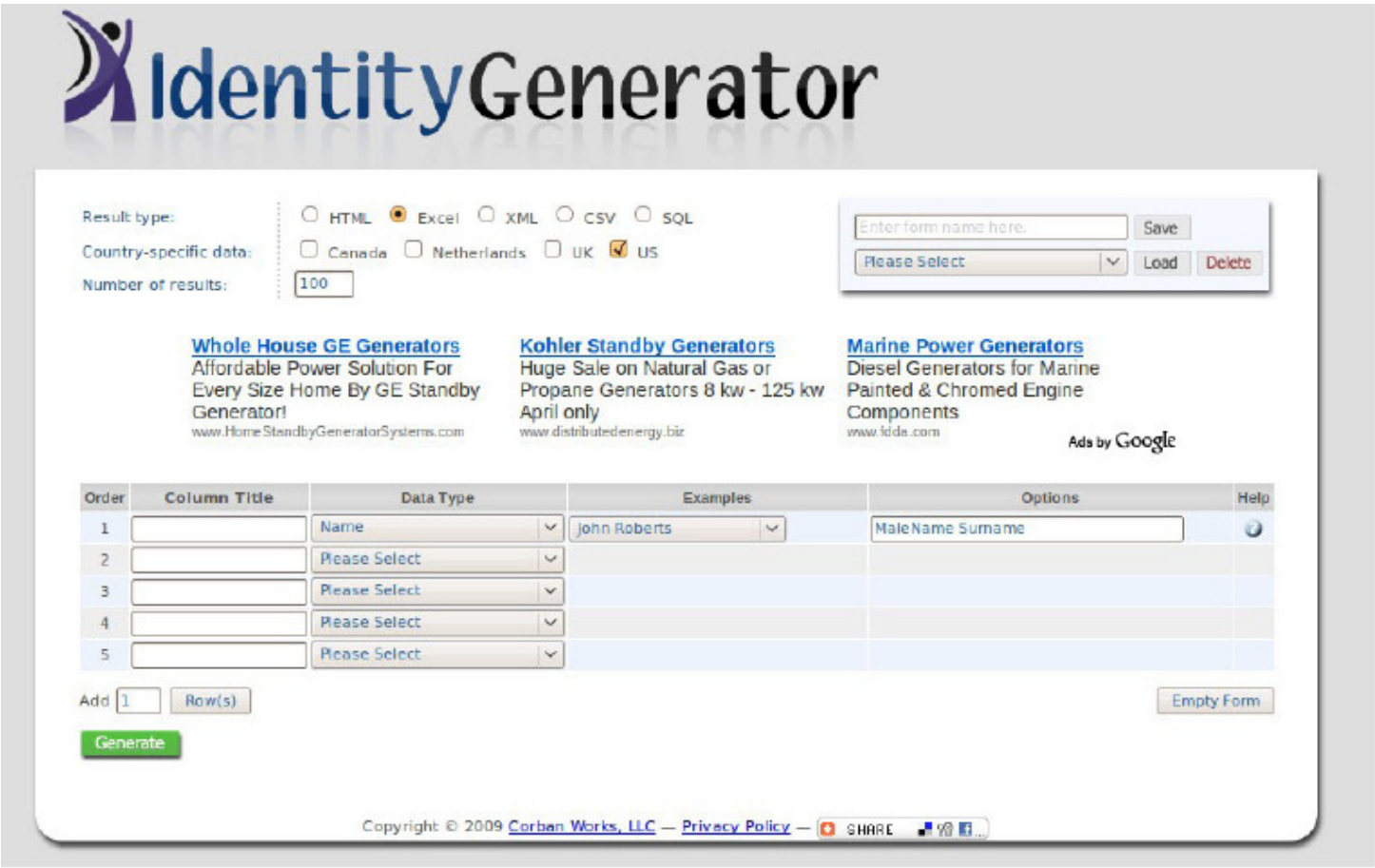
**8) From ViewAll.apx.cs file, the following SQL statement is used to get all the members from the members table:**

```
"SELECT * " +
"FROM Members ";
```

# Tools used to Generate Data:

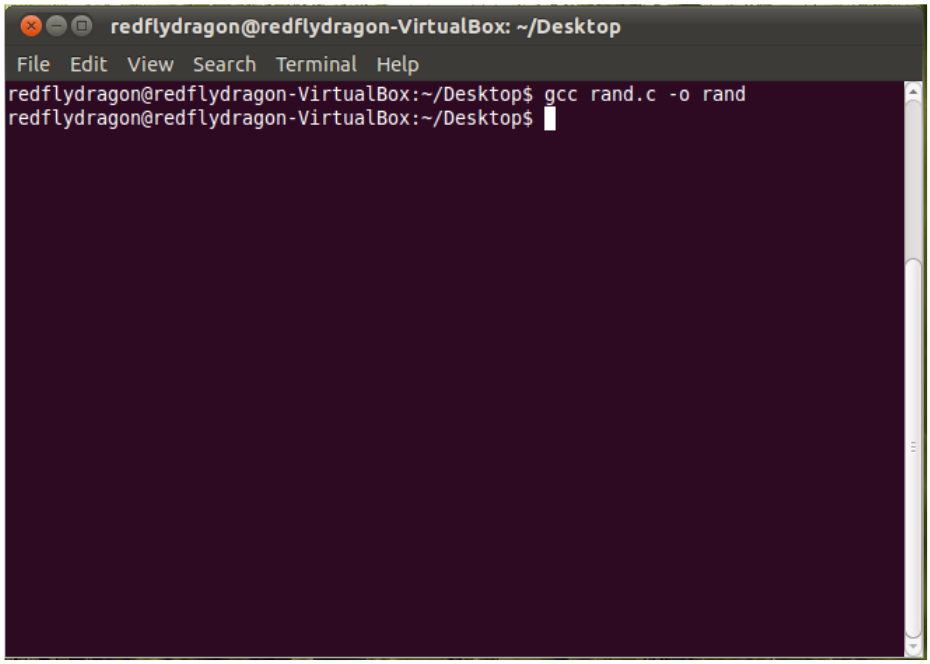
## a) Identity Generator:

The identity generator software was used to generate data like the first name, last name, id, address, city, state, zip code, and password for the MEMBERS table. This program give you the option of create the result in the fallowing formats HTML, Excel, XML, CSV, and SQL. It give you the option of specify the country-specific data that you want to create like Canada, Netherlands, UK, and US. For last it let you specify the number of results that you want to generate. The software is totally free and it can be found on the fallowing site http://www.identitygenerator.com.

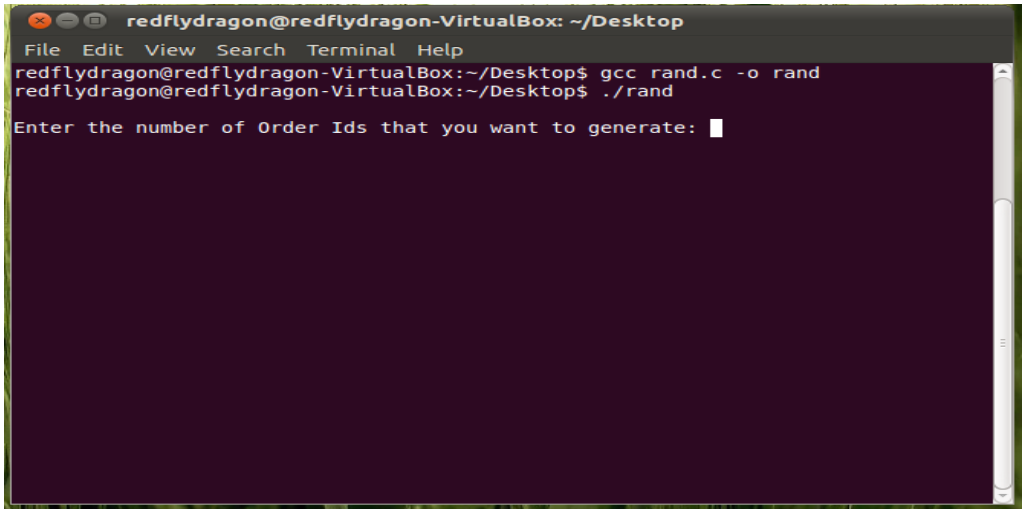



## b) rand.c

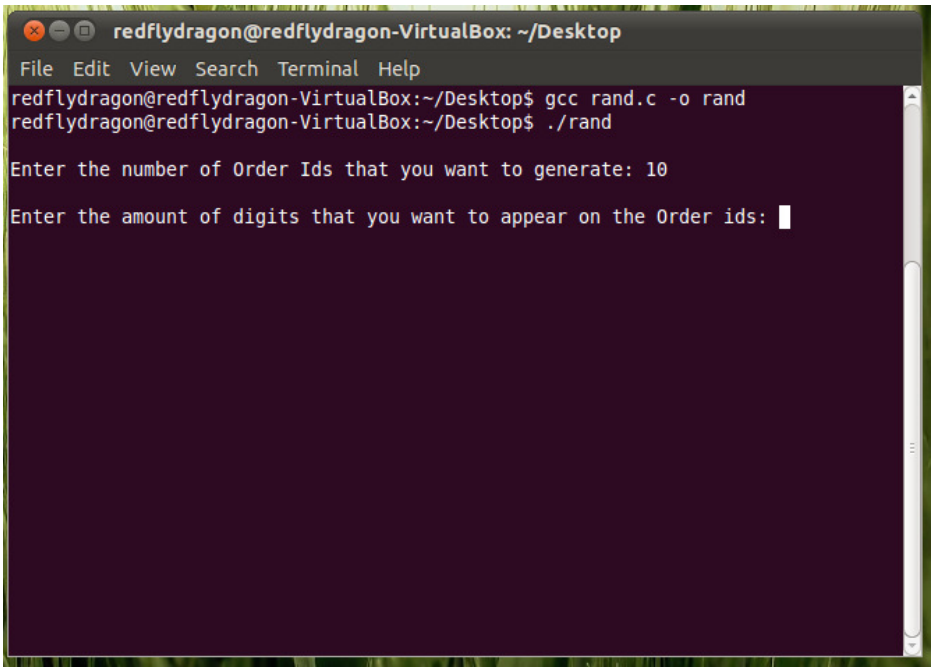
This program generates random orders number for the ORDER table. It can generate a specific number of orders numbers and the amount of digits that are specified by the user. To compile the program follow this command: “gcc rand.c –o rand”. To run the application and print the results on a text file use the following command: “./rand >>ids.txt”.



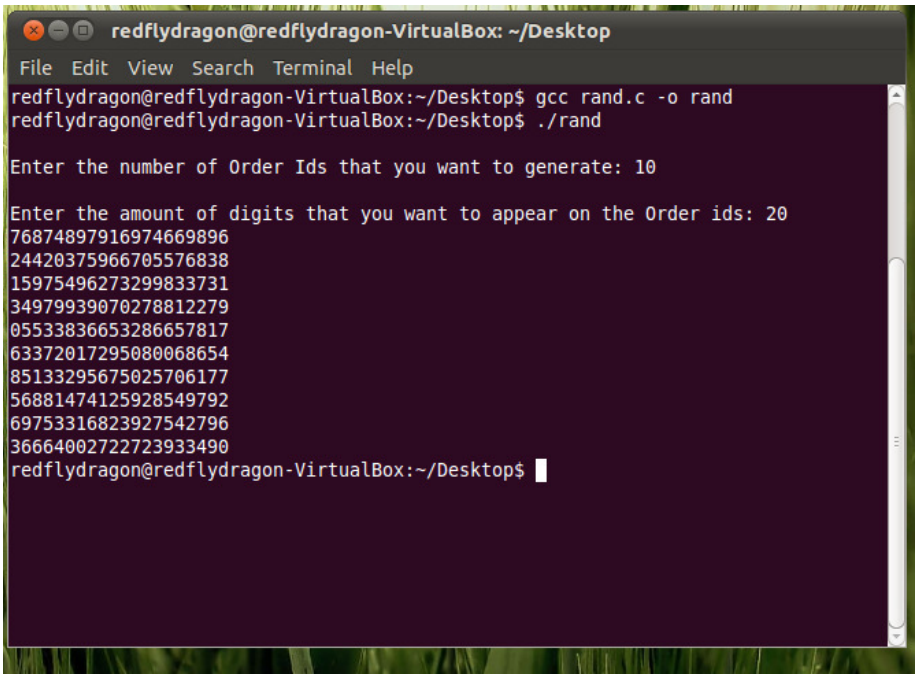

In this screenshot show how the program is compiled using the gcc compiler.


redflydragon@redflydragon-VirtualBox: ~/Desktop
File Edit View Search Terminal Help
redflydragon@redflydragon-VirtualBox:~/Desktop$ gcc rand.c -o rand
redflydragon@redflydragon-VirtualBox:~/Desktop$ ./rand
Enter the number of Order Ids that you want to generate:

This screenshot shows how to run the program after it is compiled. Immediately the program is running it will ask the user the amount of digits that each order id will have.


redflydragon@redflydragon-VirtualBox: ~/Desktop
File Edit View Search Terminal Help
redflydragon@redflydragon-VirtualBox:~/Desktop$ gcc rand.c -o rand
redflydragon@redflydragon-VirtualBox:~/Desktop$ ./rand
Enter the number of Order Ids that you want to generate: 10
Enter the amount of digits that you want to appear on the Order ids:

After entering the number of digits for the order id the program will ask to the user the amount of order of ids to generate.




This screenshot show the result of the order ids generate by the program. This result can be printed to the screen of the computer or to a text file using the following command “>> name_of_the_file”.

# Other Issues:

Since this program currently runs on a local machine, it must be loaded and run in Microsoft Visual Studio (minimum version of 2008). In addition, since this is a web application which connects to a local database, the location of the database varies from computer to computer. Since this location changes, a new connection string must be entered for SQL queries to be completed successfully. A step by step procedure to accomplish this is found below.

1. Once in Visual Studio and the project folder has been opened as a "Website" open Server Explorer under "View"
2. In the Server Explorer, under Data Connections, right-click Database.mdf
3. In the properties panel, find Connection String and copy it
4. Take the connection string and make it look like the one below with regard to quotation marks and the @ symbol :
5. `@"Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\Dan Ehlers\Desktop\final_project_update_6\App_Data\Database.mdf;Integrated Security=True;User Instance=True";`
6. Then press Ctrl+Shift+F to bring up the Find and Replace dialog box
7. Click Quick Replace and select Replace in Files
8. In the Find What box copy the connection string in step 5
9. In the Replace box, copy your connection string from the properties
10. Make sure Look In: says Entire Solution and click Replace All button at the bottom
11. Project is ready to run!